# Nonlinear Interactive Motion Control Techniques for Virtual Space Navigation

Deyang Song and Michael Norman
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
Urbana, IL 61801

**Abstract**

For virtual reality scientific visualization applications, the traditional navigational analogue of moving in a room (the "immersion" technique) is not adequate. Moving out of the simulation box and getting a panoramic view of the visualization is essential. In this way the sense of immersion is augmented by the ability of overviewing. Visualization applications also require that interactions on a global scale remain possible at any time. We present in this paper nonlinear motion control techniques that allow fast and intuitive control of the viewpoint as well as the hand position within a virtual workspace. They allow the users to quickly move in and out of the simulation box, and at the same time interact with the simulation using the hand device no matter where the viewpoint is located. The principle is to divide the working range of a physical input device into several parts and use different mapping functions to map the parameters of the device into virtual space. Providing a consistent cognitive model and maintaining smooth transitions between subspaces are the basic requirements for this type of techniques. Our proposed techniques are natural to users working with 3D input and output devices. They are simple and fast so they don't add additional time lags to many VR applications that are already tied by system response time. The application of these nonlinear motion control techniques in the Cosmic Explorer, a virtual reality visualization system in cosmology, shows that users adapt to these techniques instantly and their responses are rather positive.

## 1 Introduction

Virtual Reality (VR) systems are ideal interfaces to scientific visualizations. By using stereoscopic displays [5, 11, 15] and 3D input devices [2, 14, 17], scientists can see their data in 3D directly. 3D input and output devices are significantly different from traditional 2D devices. For one thing, users now use multiple devices to interact with the computer. Another thing is that the devices work in larger ranges than, say, mice. These new features open new possibilities but also raise new questions. New interactive techniques need to be developed that are specifically suited to the nature of the VR systems.

There are two fundamental properties of a VR system: immersion, the illusion of being inside a computer-generated scene, and navigation, the ability to move around inside the scene [13]. In this paper we investigate navigational techniques that not only give users the illusion of immersion, but also give users the ability of overviewing. For a scientist not only wants to be able to inspect the data closely but also wants to be able to pull out and see the whole picture. We will present two motion control techniques for viewpoint control and for hand motion control.

A popular VR interface model, named the cyberspace model [16], uses the "walking" metaphor. In this model the user walks in a room, sees the scene in stereo and gets the

0-7803-1363-1/93 $3.00 

feeling of being surrounded by the data [1, 2, 16]. Other interface models usually involve using the hand to control a physical device that in turn controls the viewpoint either directly [10] or through a virtual device [4, 8, 16]. Following Mackinlay's classifications [10], the cyberspace model falls in the category of general movement, meaning the user controls the positioning of the viewing device at any point and doesn't follow any calculated path. Our motion control techniques extend the simple cyberspace model by decoupling virtual space movement from physical space movement using nonlinear mapping functions, the result of which is a unified approach to satisfy the needs of both immersion and overviewing.

In section 2, we review related work on motion control techniques. In sections 3 and 4, we present our viewpoint and hand motion mapping techniques respectively. The last section contains one application of our techniques and remarks on future work.

## 2 Related work

Using decoupling techniques to control viewpoint movement through virtual devices has been reported in many literatures [4, 7, 8]. Using viewing devices to directly control viewpoint movement is relatively new [1, 5, 11], although the idea was proposed rather early [15].

There are generally two approaches to control the viewpoint. One is through object manipulation, another is through viewpoint specification. The techniques that manipulate objects [4, 14] generally do not apply in scientific visualization environments, because the graphical objects are not generated from a mathematical model and thus hard to identify. The only thing that can be manipulated may be the "root object", which is the entire data domain [16].

In [10] Mackinlay proposed an exponential function for targeted movement control. The target is defined as the intersection of the moving direction and the nearest object surface. Mackinlay's technique allows the user to move rapidly close to a *point of interest* (POI) and get a fine motion control near the POI. This technique is good for targeted movement but less effective for general viewpoint movement. Because in general movement, a user constantly changes his POI. Requiring the user to specify a POI every time before he starts moving is distracting. Our viewpoint control techniques provides a solution to the problem of general exploratory movement. Additionally, our hand mapping algorithm solves the problem of specifying an arbitrary POI within the virtual workspace. The intersection method mentioned in [10] will not allow a user to move along a line where no intersection can be found, or in a space where no surface exists.

The problem of being unable to reach any point in a scene with a hand device using the simple cyberspace model has also been reported by Ware in [16]. In Ware's report the user has to physically walk to the point where he wants to position the viewpoint. Our hand motion control technique will overcome this problem.

## 3 Viewpoint movement

The viewpoint motion control involves mapping physical space coordinate $V(x_v, y_v, z_v)$, the location of the viewing device (e.g. BOOM), into virtual space coordinate $V'(x_{v'}, y_{v'}, z_{v'})$, the viewpoint, by the following function:

$$V' = f(d)V \tag{1}$$

where $d = \mid V \mid = \sqrt{x_v^2 + y_v^2 + z_v^2}$ represents the physical distance of the viewing device from its origin ($0 \le d \le d_{max}$), and $f$ is defined as:

$$f(d) = \begin{cases} \dfrac{D_0}{d_0} & \text{if } d < d_0 \\ \dfrac{D_0 e^{(d-d_0)/d_0}}{d} & \text{otherwise} \end{cases} \tag{2}$$

The meanings of $d_0$ and $D_0$ will be explained shortly.

Let $d' = \mid V' \mid = \sqrt{x_{v'}^2 + y_{v'}^2 + z_{v'}^2}$. Equations (1) and (2) yield:

$$d' \equiv F(d) = f(d)d = \begin{cases} \dfrac{D_0 d}{d_0} & \text{if } d < d_0 \\ D_0 e^{(d-d_0)/d_0} & \text{otherwise} \end{cases} \tag{3}$$

Figure 1 shows the function $F$ with $D_0 = 1.0$, $d_{max} = 1.0$, and $d_0$ of different values. Note that $F$ is composed of two functions, a linear function $F_1(d) = D_0 d/d_0$ and a nonlinear function $F_2(d) = D_0 e^{(d-d_0)/d_0}$. To give the viewer a sense of smooth motion in the virtual space the only requirement is that $F_1$ and $F_2$ have $C^1$ continuity at the connecting point $(d_0, D_0)$. It is easy to prove that the two functions satisfy the $C^1$ continuity condition.

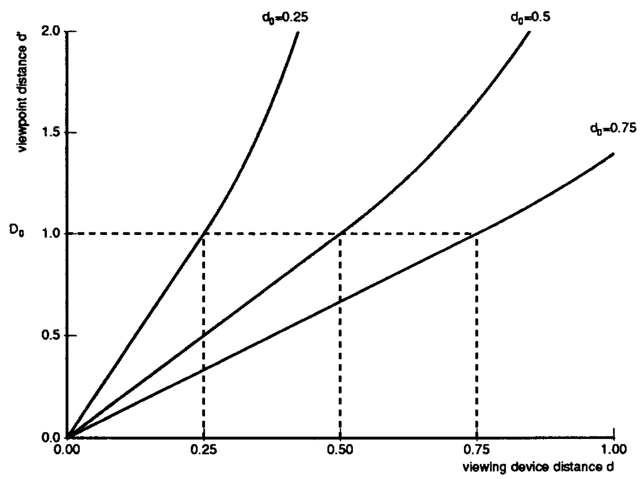


Figure 1: View mapping function $F : d \mapsto d'$, with $D_0 = 1.0$, $d_{max} = 1.0$, and $d_0 = 0.25$, $0.5$, and $0.75$ respectively.

Function $F$ effectively divides the virtual space into two parts, separated by a sphere $\sqrt{x_{v'}^2 + y_{v'}^2 + z_{v'}^2} = D_0$ with radius $D_0$. We call this sphere and the space inside it the *Domain Of Interest* (DOI). The physical subspace corresponding to DOI is called *Linear Viewing Space* (LVS), and it is a sphere defined by $\sqrt{x_v^2 + y_v^2 + z_v^2} = d_0$. Equal amount of movement inside the LVS translates into equal amount of movement in the DOI. Thus close-up inspections can be performed within this subspace. This is the traditional "immersion" technique. Outside the LVS, the same amount of physical movement translates to larger virtual space movement when the viewer gets further away from the center of the physical space. The $C^1$ continuity guarantees that the transition from one subspace to another is smooth.

The $d_0$ indicates how much of the working range of the viewing device is used for linear motion control inside the DOI and how much is used for exponential motion control outside the DOI. We can see from Figure 1 that in order to increase the speed of recession from the center of the DOI, we need to sacrifice the range of linear space devoted to fine control inside DOI. When $d_0 = d_{max}$, we get a totally immersive style VR system. The principle of devoting part of the working space of the physical device for special purpose will also be applied to hand motion control.

## 4 Hand movement

Our philosophy behind the hand motion control is that when a user is interacting with a visualization system, he should be able to reach any point within the data domain regardless of where he is standing. Note that in our hand motion control algorithm described below, two interactive devices, both the viewing device and the hand device, are involved. The hand device always moves with the viewing device, since the latter is attached to the user's head and the former to the user's hand. To clarify further discussions, we make the following assumptions:

1. the data domain does not change its position;
2. the center of the data domain coincides with the center of the DOI, as well as the origin of the coordinates of the virtual workspace;
3. the DOI is a sphere.

The existence of a DOI implies that when the viewpoint is outside the DOI, the entire data domain lies within a viewing cone defined by the viewpoint and the DOI. We are only interested in (virtual) hand motions within the DOI. For that purpose we define a sphere in the physical space centered around a point near the viewing device. Hand movement within that sphere is mapped to the DOI. Hand movement outside the sphere will not be considered as interactions with the visualization environment. This has the benefit of filtering out casual gestures that a user inevitably makes from time to time. We call this sphere in the physical space the *Hand Working Space* (HWS).

Some notations that will be used are explained below. For illustrations see Figure 2.

$H$ is the position of the hand device (e.g. DataGlove) in physical space;
$H'$ is the hand position in virtual space;
$V$ is the position of the viewing device (e.g. BOOM) in physical space;
$V'$ is the position of the viewpoint in virtual space;
$O$ is the center of the HWS;
$O'$ is the origin of the virtual space, as well as the center of the DOI;
$O_p$ is the origin of the physical space;

The hand mapping algorithm maps $H$ to $H'$:

**Algorithm** (*hand mapping*)

1. Get the position of $V(x_v, y_v, z_v)$;
2. Determine the position of $O(x_o, y_o, z_o)$;
3. Get the position of $H(x_h, y_h, z_h)$;
4. Calculate the relative position of $H$ w.r.t. $O$;
5. Map $H$ to $H'(x_{h'}, y_{h'}, z_{h'})$. ∎

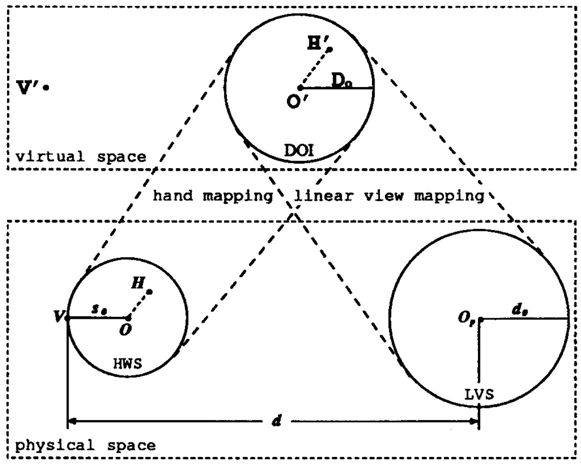


Figure 2: Illustration of hand mapping.

Note that steps 1–4 are equivalent to moving the Polhemus source, in the case of a DataGlove, to point $O$. The central idea of this algorithm is to calculate where $O$ should be located.

$V(x_v, y_v, z_v)$ in step 1 is returned by the viewing device. $O(x_o, y_o, z_o)$ is calculated by the following formula:

$$O = g(d)V \tag{4}$$

where $d$ is defined in Equation 1, and

$$g(d) = \begin{cases} \dfrac{d_0 - s_0}{d_0} & \text{if } d < d_0 \\ \dfrac{d - s_0}{d} & \text{otherwise} \end{cases} \tag{5}$$

where $s_0$ is the radius of the sphere of the HWS. It is also the distance from the location of the viewing device to the center of the HWS. In practice, $s_0 =$ half the arm length is a good choice. The $g(d)$ is derived based on the following idea: when the viewpoint is outside the DOI, the user should be able to have a global control; when inside the DOI, the user gets the sense of immersion. In other words, if the viewpoint is outside the DOI, the HWS is limited to one side of the viewpoint, centered along the line $VO_p$ (see Figure 2). If the viewpoint is inside the DOI, the HWS should surround the the point $V$ just as the DOI surrounds the viewpoint $V'$, which is just what the user feels (see Figure 3).

The relative position of $H$ w.r.t. $O$ is $H - O$. To map $H - O$ to $H'(x_{h'}, y_{h'}, z_{h'})$ we use the following formula:

$$H' = D_0(H - O)/s_0 \tag{6}$$

Putting steps 1–4 of the above algorithm together, we get:

$$H' = D_0(H - g(\mid V \mid)V)/s_0 \tag{7}$$

## 5 Concluding remarks

The motion control techniques have been applied to our VR application, the Cosmic Explorer, that visualizes galaxy formation in the universe in space and time as computed by numerical cosmology simulations [3]. The system uses the DataGlove and BOOM as input and output

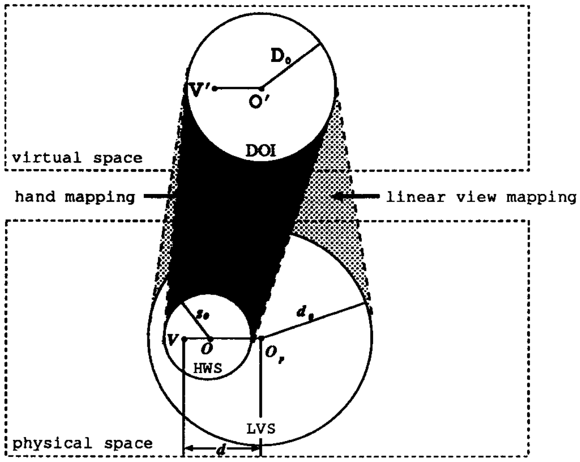


Figure 3: Hand mapping: determine the point $O$ when $d < d_0$. The hand position $H$ and $H'$ are not drawn in this diagram.

devices. The visualization shows large structures such as clusters, super clusters, filaments, voids, etc, that have been observed in the real Universe [6].

In our Cosmic Explorer, about half of the BOOM's physical space is used for controlling viewpoint movement outside the data box, and another half used for fine control inside the box (DOI). We found this choice quite satisfying. The principle of dividing up the working range of a physical device and devoting each part for one special purpose is applicable to other types of virtual reality applications besides scientific visualization.

Our motion control strategies correspond to an extremely simple cognitive model. The user only moves his head around without thinking of what metaphors is being used. Although in other complex spaces this is not sufficient, it works very well for our application. And it overcomes problems associated with simple cyberspace models. In addition, the nonmodal property of our techniques avoids getting users attention shifted from interacting with the data to, say, changing the speed scale.

The accuracy of the viewing device and the hand device is crucial, because we are using only part of its effective working range. The DataGlove we are using now does not provide position values accurate enough due to the sensitivity of the Polhemus sensor to the electromagnetic field. When we only use part of its working range for the HWS, the motion of the 3D cursor that represents the hand position in virtual space becomes jerky. Positional data filtering may be needed to reduce the noise [9]. But we suspect that filtering would introduce more lag times thus decrease the degree of interactivity [14].

Future extensions include using multiple control functions. This will be necessary in a more complex information space where there are many DOIs.

## Acknowledgement

We would like to thank Steve Bryson from NASA Ames for his generous support and valuable comments during the development of the Cosmic Explorer. In fact, our BOOM and DataGlove code was based on his Virtual Windtunnel program [2]. We would also like to thank Michael McNeill at NCSA for his technical support for the Virtual Reality Lab. This

work was partially supported by the NSF under grant number AST9012353 and utilized Silicon Graphics 4D/440 at the National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign.

## References

[1] Brooks, F. P. Jr., "Walkthrough – A Dynamic Graphics System for Simulating Virtual Buildings," *Proc. 1986 ACM Workshop on Interactive Graphics*, October 1986, pp. 9–21.

[2] Bryson, S. and C. Levit, "A Virtual Environment for the Exploration of Three Dimensional Steady Flows," In *Stereoscopic Displays and Applications II*, J. O. Merritt, S. S. Fisher, Editors, Proc. SPIE 1457, 1991, pp. 161–168.

[3] Cen, R. Y., et al., "A Hydrodynamic Approach to Cosmology – Texture-seeded Cold Dark Matter and Hot Dark Matter Cosmogonies," *Astrophysical Journal*, Vol. 383, No. 1, December 1991, pp. 1–18.

[4] Chen, M., et al., "A Study in Interactive 3-D Rotation Using 2-D Control Devices," *Proc. SIGGRAPH'88*, August 1988, pp. 121–129.

[5] Chung, J. C., et al., "Exploring Virtual Worlds with Head-Mounted Displays", In *Three-Dimensional Visualization and Display Technologies*, W. E. Robbins, S. S. Fisher, Editors, Proc. SPIE 1083, 1989, pp. 35–41.

[6] Geller, M. J. and J. P. Huchra, "Mapping the Universe," *Science*, Vol. 246, November 1989, pp. 897–903.

[7] Glassner, A. S., "A Two-Dimensional View Controller," *ACM Trans. Graphics*, Vol. 9, No. 1, January 1990, pp. 128–141.

[8] Hultquist, J., "A Virtual Trackball," In *Graphics Gems*, A. S. Glassner (ed.), Academic Press, Inc., San Diego, California, 1990, pp. 462–463.

[9] Liang, J., et al., "On Temporal-Spatial Realism in the Virtual Reality Environment," *UIST'91*, November 1991, pp. 19–25.

[10] Mackinlay, J. D., et al., "Rapid Controlled Movement Through a Virtual 3D Workspace," *Proc. SIGGRAPH'90*, August 1990, pp. 171–176.

[11] McDowall, I.E., et al., "Implementation and Integration of a Counterbalanced CRT-based Stereoscopic Display for Interactive Viewpoint Control in Virtual Environment Applications," In *Stereoscopic Displays and Applications*, J. O. Merritt, S. S. Fisher, Editors, Proc. SPIE 1256, 1990, pp. 136–146.

[12] McKenna, Michael, "Interactive Viewpoint Control and Three-Dimensional Operations," *Proc. 1992 Symposium on Interactive 3D Graphics*, March 1992, pp. 53–56.

[13] Rheingold, H., *Virtual Reality*, Summit Books, New York, 1991.

[14] Sturman, David J., et al., "Hands-on Interaction with Virtual Environments," *UIST'89*, November 1989, pp. 19–24.

[15] Sutherland, I. E., "A Head-Mounted Three Dimensional Display," *1968 Fall Joint Computer Conference*, Vol. 33, 1968, pp. 757–764.

[16] Ware, C. and S. Osborne, "Exploration and Virtual Camera Control in Virtual Three Dimensional Environments," *Computer Graphics*, Vol. 24, No. 2, March 1990, pp. 175–183.

[17] Zimmerman, T. G., et al., "A Hand Gesture Interface Device," *Proc. CHI'87*, 1987, pp. 189–192.